

Ruby Palabras Reservadas

`__ENCODING__` : codifica el script del archivo actual.

`__LINE__` : el número de línea del archivo actual.

`__FILE__` : la ubicación del archivo.

`BEGIN` : indica lo primero que se va a ejecutar en el archivo.

`END` : indica lo último que se ejecuta en el código en el archivo.

`alias` : crea un alias entre 2 métodos.

`and` : operador lógico “Y” con menor precedencia que `&&`

`begin` : marca el inicio de un bloque

`break` : detiene la ejecución antes en un bloque de código.

`case` : inicializa la expresión case.

`class` : para inicializar una clase.

`def` : define una función

`defined?` : regresa un string describiendo su argumento

`do` : empieza un bloque.

`else` : maneja la otra respuesta en un `if` o `unless`.

`elsif` : condición alterna de un `if`.

`end` : finaliza la sintaxis de un bloque.

`ensure` : se asegura de correr un bloque de código si una excepción es levantada.

`False`: valor booleano false.

`for` : ciclo mayormente utilizado para recorrer objetos iterables.

`If` : para inicializar una expresión `if` que nos sirve para escoger que código ejecutar.

`in` : para indicar en que objeto iterable va a iterar el ciclo `for`.

`module` : para inicializar un modulo.

`next` : se salta lo siguiente en un bloque.

`nil` : para indicar que un valor esta vacío o desconocido.

`not` : invierte la expresión booleana y cuenta con menor precedencia que `!`

`or` : operador lógico “o” y con menor precedencia que `||`

`redo` : reinicia la ejecución en el bloque actual

`rescue` : inicia la sección de una excepción en un bloque

`retry` : reintenta el bloque de excepción

`return` : para salirse de la ejecución de un método.

`self` : el objeto del método al cual está actualmente atado.

`super` : llama al método actual en una super clase

`then` : indica el final en una estructura de control

`true` : valor booleano verdadero

undef : previene que se ejecute un método

unless : es lo contrario a la estructura de control if.

until : ejecuta un ciclo hasta una sentencia sea verdadera.

when : indica una condición cuando se usa la estructura de control case.

while : ejecuta un ciclo mientras una condición se cumpla.

yield : ejecuta un bloque enviado a una función